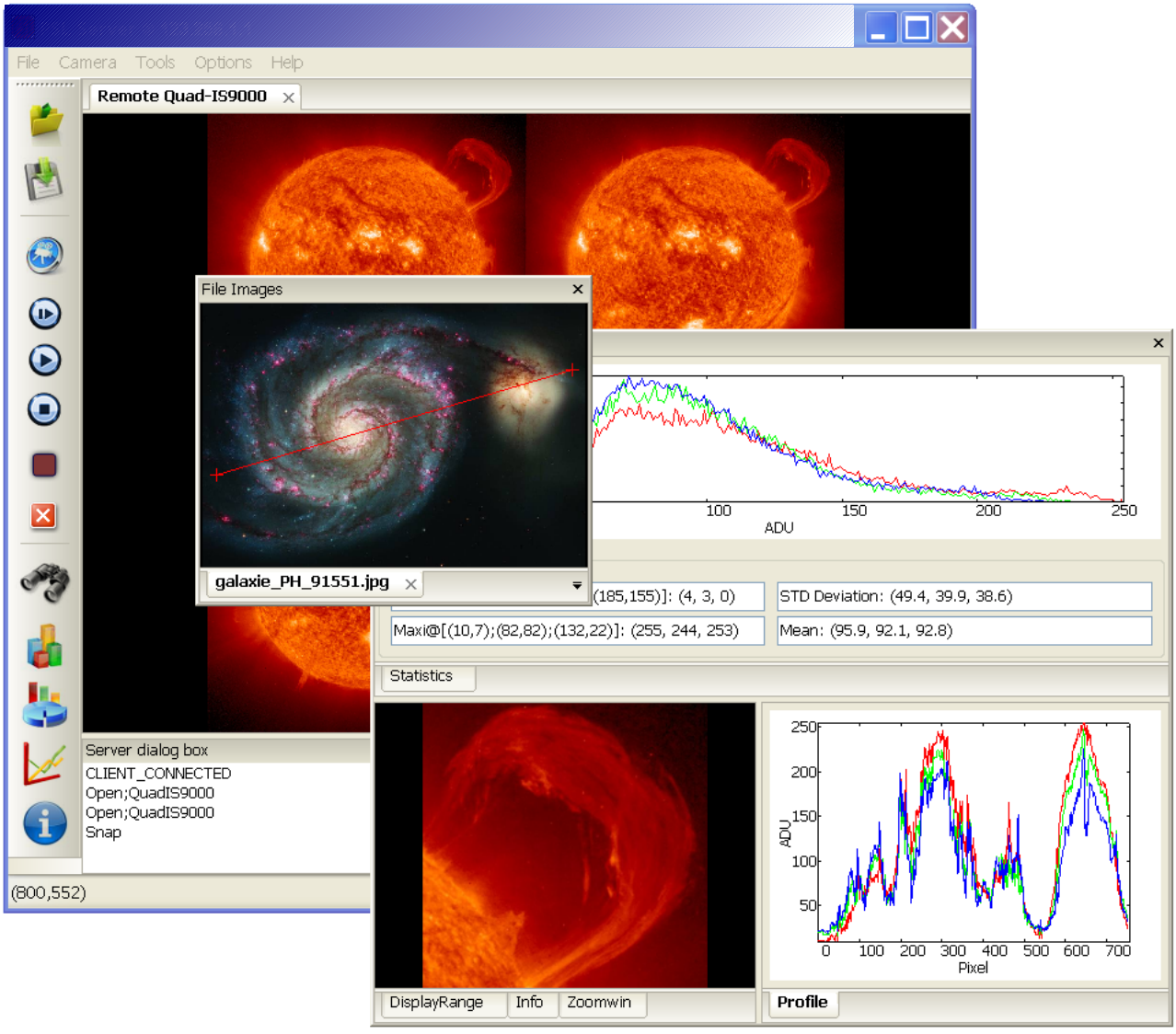
Photonic Science

Scientific Detector Systems

SOFTWARE MANUAL



INDEX

1. Installation

1) Frame grabber

• Option 1: Install IFC 5.9 software

- insert the IFC 5.9 CD
- copy *IFC59.zip* to a directory of choice on hard drive, and unzip (extract files)
- run *"setup.exe"*
- allow IFC 5.9 to be installed in the default folder (*"c:\Ifc59"*)
- choose *"Typical setup"*
- select the *"PC2-Camlink"* framegrabber from the 2nd page of the *"Select Components"* dialog
- allow installation to proceed, and answer *"NO"* to check for an IFC service pack update
- install the IFC service pack contained within *"IFC59SP5a.zip"* (supplied)
- now SHUT DOWN THE COMPUTER

• Install the PC2-Camlink framegrabber in any spare PCI slot and and then restart the PC. Windows should detect the new hardware and offer to load the appropriate software (drivers). Allow it to do so.

IMPORTANT: Go to *"Start -> Progs -> IFC 5.9.0 -> Tools -> Set Board Com Port"* and hit the *"Reset Name"* button. (The default port name for the PC2-Camlink should remain set at *"PC2-Camlink_1_Serial_0"* when you do this). You will be prompted to restart the PC at this point. If you do not perform these steps then the camlink serial interface will be incorrectly configured, and the driver will not be able to run.

• Option 2: Install Multicam v4.5 software

- insert the Euresys Windows CD and click on Install
- choose *"Run-time"* from Setup Type and *"Multicam"* from Products to Install
- the default installation folder is acceptable
- click *"Next"* and *"Next"* again to start installation proper
- when the installer software has finished, allow the system to be rebooted

Note: if any other version of the Multicam software has already been installed on the host PC, then this should be removed first, and the machine then rebooted before proceeding to install Multicam v4.5. To check for and remove existing versions of Multicam, go into: *"Control Panel->Add/Remove Programs" and search for "Euresys Multicam"*

• Shut PC down once again, and install the Euresys Grablink framegrabber in a free PCI slot. Start the PC and, when windows detects the new hardware, allow it to search for an appropriate driver. When finished, go to *"Start->Control Panel->System->Hardware->Device Manager"* and confirm that *"Grablink"* now exists under *"EureCard"*

• Option 3: Install IPORT software

- Install PRO/1000 GT gigabit network card and then start the PC. Windows identifies the card and asks whether it should *"connect to Windows Update to search for software?"* – answer *"No, not this time"*. When prompted, insert the CD *"Intel PRO Network Connections"* into the PC. When asked *"What do you want the wizard to do?"* reply *"Install the software automatically"*. Windows should then load the necessary drivers from the CD, and eventually announce *"The Wizard has finished installing the software for Intel PRO/1000 GT Desktop Adapter"*. Intel PRO/1000 GT adapter should also be shown in Device Manager under Network Adapters.

• ***** IMPORTANT - REBOOT THE PC *****

• Connect the IPORT engine (black box) to the Install PRO/1000 GT network card with the network cable

• Run the *"iPort_IPEngineSDK_2.3.3.740.exe"* from the supplied CD, and select BOTH the eBUS Driver Suite and the iPort Vision Suite software. Following installation of the eBUS Driver Suite, you may be prompted to install Microsoft Fix 9213*37 in order to*

make the driver software more stable – this can be done by running "WindowsXPKB921337-x86-ENU.exe", which you will also find on the CD.
Alternatively, for IPORT engines with 3 digit firmware versions (5.0.0+) please use the updated IPORT software contained within *"iport_IPEngineSDK_3.00.00.1089.zip"*, and run *"setup.exe"* from the extracted files. This latest IPORT SDK may require the user to first install Microsoft .NET Framework Version 2.0 (available as a download).

• Next we must configure the network connection for use with the IPORT device. Refer to the *"eBus quick start guide"* to be found under *"Programs->Pleora Technologies Inc->eBus Driver Suite->Documentation"*. The eBus driver syphons of IPORT packets and sends them straight to IPORT applications. Several driver options are available, but because we have installed a PRO/1000 GT card the *"eBUS Optimal Driver"* should be selected. This "optimal driver" actually replaces the driver originally supplied with the PRO/1000 GT card, and maximises bandwidth and minimises CPU usage.
To install the "eBus Optimal Driver":
o go to *"Start->Programs->Pleora Technologies Inc->eBus Driver Suite"* and run the *"Driver Installation Tool "*
o select the PRO/1000 GT card and hit *"Configure"*
o choose the *"Optimal Driver"*
o *** **IMPORTANT - REBOOT THE PC** ***
[Please note that with the later IPEngineSDK_3.00 software, the Driver Installation Tool dialog may appear automatically.]

• It remains to set up the IP address for the network card. See the "quick start guide" for full details, but here we are going to setup a static IP address by performing the following:
o *"Start->Control panel->Network Connections"*
o Select *"Firewalled Ethernet Bus Network Interface"* (i.e. the new network card) and right click mouse and select properties
o Make sure only *'Internet Protocol'*, *'Show Icon ....'* and *'Notify me ....'* are ticked.
o Select *'Internet Protocol (TCP/IP)'* and click *'properties'*.
Use the following IP address:
- *IP address 192.168.101.001* (for example)
- *Subnet mask 255.255.255.0*
- *Default gateway [leave blank]*
Click OK and OK again to close the dialog boxes

• It is necessary at this point to power cycle (off – on) the IPORT black box at this point in order to make it visible on the network

2) Software

To install the software on your computer, just copy the application main folder to: "C:\Program Files\"
To start the application double click on *"..**SoftwareSXxx.exe***".
You can create a shortcut for your desktop by right clicking on the executable file.

3) Camera files

In order to make your camera visible in the software menu, a folder with your camera name has been created in the main application directory.
This camera_folder contains all the files required for your camera and includes:
– the camera control DLL file (! only one DLL file must exist at this location/directory level !)
– the ***"PSL_camera_files"*** folder (contains all the camera correction files)

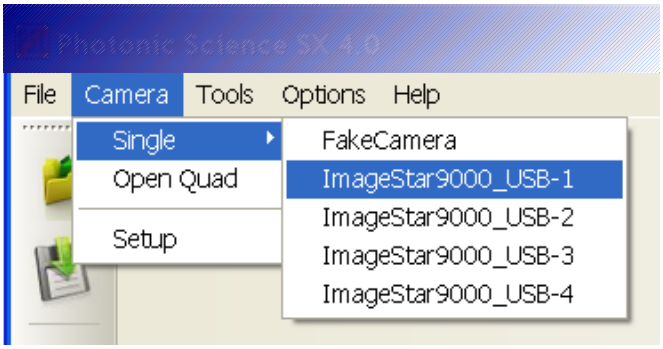
If you need to re-create this folder, you can find the list of the valid camera_folder names in the "*camtypes.dat"* file located in the main application directory.

IMPORTANT NOTE:
The camera correction files included in *"PSL_camera_files"* are specific to your camera and should not be modified, exchange or removed. Using correction files not designed for your camera, will affect the camera imaging quality and the hardware stability. A safe backup of the control DLL file and the *"PSL_camera_files"* folder can be found on the CD delivered with your camera.

2. Select a camera

1) Single camera

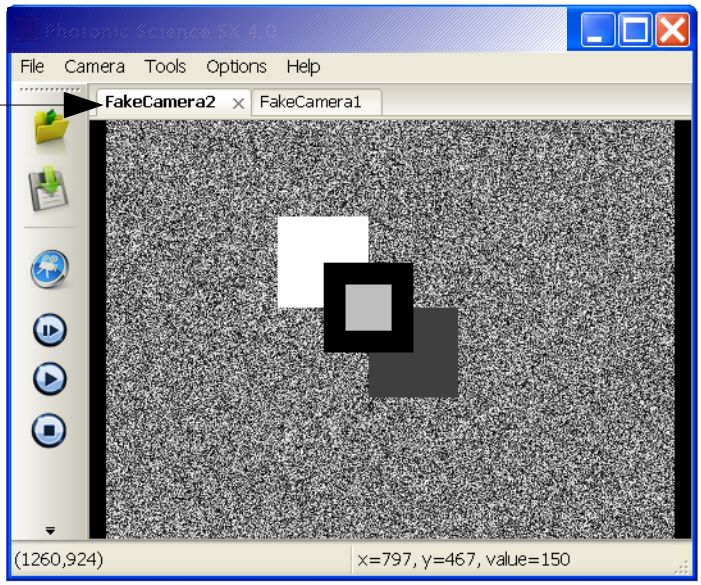
In the software menu bar, click on *"Camera->Single->..."* and select your camera.



This application is able to manage multiple 'Single' cameras. The different cameras are stored as note-book pages. Select the active camera by clicking on the page Tab. You can also use the little arrow on the top right of the note-book.

Two camera displays stored as note book pages:

Click on a page Tab to select the active display

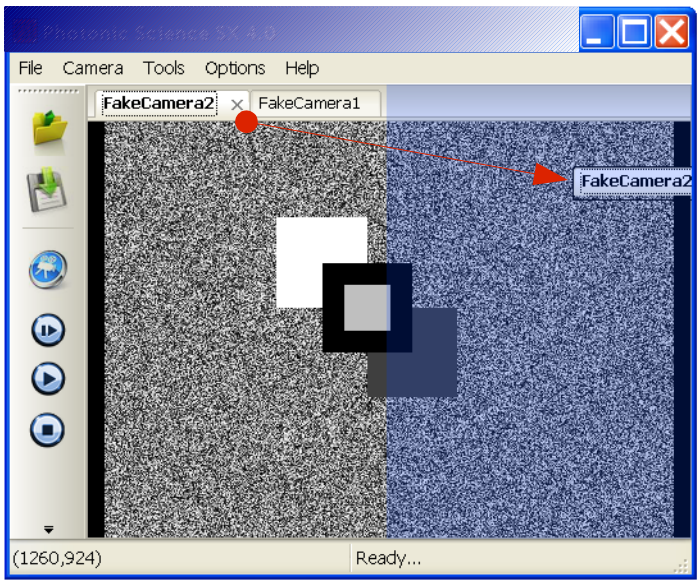


You can display several cameras at the same time. To do so, grab the page/Tab (hold left click) and drag it to one of the note book borders. A valid region is highlighted in blue. When it turns blue, be sure to release the left button when the mouse cursor is close enough to the border line.

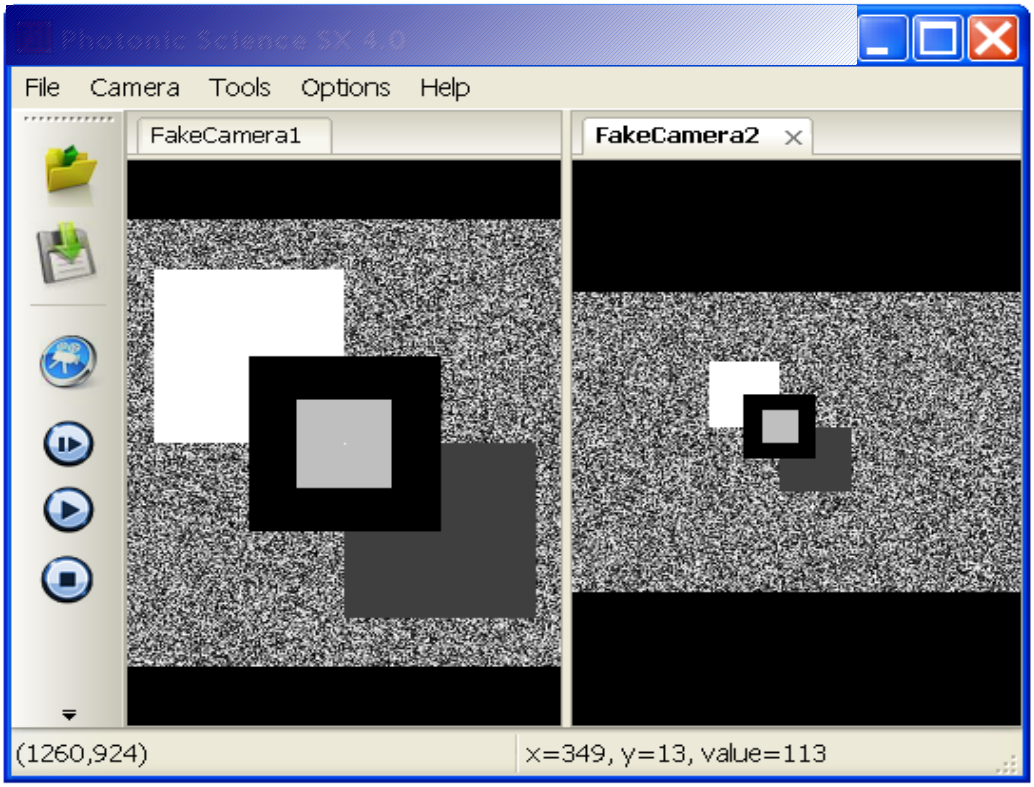
Drag and drop

Grab the page Tab (hold left click) then drag the Tab to a note-book border.

When it turns blue, drop the Tab (release left click)

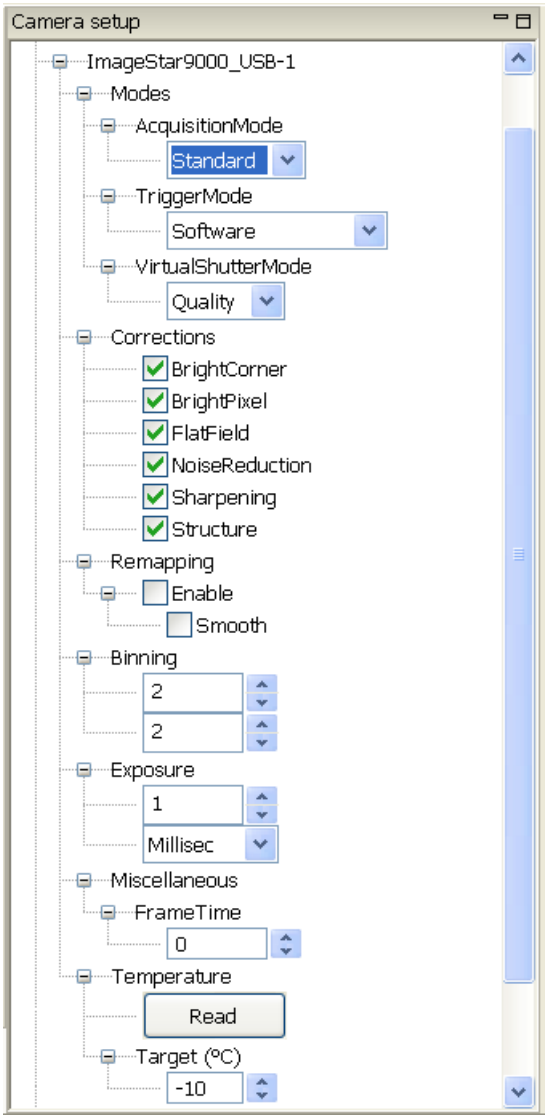


Multi display



3. Running the camera

1) Camera setup



The *"Camera Setup"* panel allow you to change various camera options. Options are described below.

Options are stored in different categories as you can see on the picture on the left. The tree root is the camera name.

Depending of your camera type, the available options may be different of what you can see here.

When changing an option value, the command is immediately sent to the camera if this one is not in an acquisition process, else the command is transmitted just after the current image acquisition.

Modes:

- **AcquisitionMode**:
  Standard is the usual acquisition mode, a new image is displayed on top of the previous.
  Average mode will display the average of the acquired images.

- **TriggerMode**: Select between the various modes of triggering.
  The camera includes an external trigger feature that allows the camera's exposure to be synchronised with an external event. Two forms of this feature are available: a hardware trigger, and a software trigger. The hardware trigger input is an optional feature, whereas the software trigger facility is always present.
  See your camera documentation for more details on available trigger modes.

Corrections:

Note: Some corrections can be enabled only if the corresponding files have been previously installed and loaded correctly during camera initialisation. Those files have to be in the *"PSL_camera_files"* folder corresponding to your camera (such files are provided with the camera, see section 1.2).

- **BrightCorner**:
  At long exposure times the dark current in this camera is spatially non-uniform, with a

higher value along the top edge of the image. Bright edge subtraction removes this dark current together with its non-uniformity, thereby giving an image with practically zero dark current.

- **BrightPixel**:
  Enable this correction to reduce the number of bright pixels (individual high dark current pixels). There are a small proportion of isolated individual pixels with higher dark current, which can appear as bright pixels at long exposure times. *"Bright pixel"* automatically subtracts these bright pixels from the image as part of the acquisition process.

- **FlatField**:
  Wide aperture lenses and fibre-optics can introduce shading across an image, resulting in reduced intensity at the edges of the image compared to the centre and in small scale patterning from fibre optic structure. Flat field correction restores the intensity in the shaded areas and removes any patterning to compensate for such effects, and produces an image of very high spatial uniformity. When enabled, this function divides the acquired image by a stored flat field image, on a pixel-by-pixel basis, as part of the acquisition process. This correction supports binning, sub-area, and gain operations. For best results, always enable *"Offset"* correction when using flat field correction.

Remapping:
Enable this option to remove the effect of fibre-optic distortion by automatically resampling the raw image. Enable ***"Smooth"*** to use sub-pixel interpolation of the original image, which eliminates any aliasing effects introduced by the resampling. Enable ***"Clip"*** to exclude edge irregularities caused by the remapping process, and to present a cleaner remapped image, particularly when sub-area operation is employed.

Binning:
On-chip binning is the process of summing the signal from neighbouring pixels prior to the readout of this signal from the CCD. It has the effect of increasing the signal amplitude (by a factor equal to the number of pixels being binned), whilst making little change to the random noise amplitude. The signal-to-noise ratio of a signal therefore improves by the binning factor. Binning reduces resolution by the binning factor in both X and Y directions. Binning in the Y-direction also reduces the time taken to read out the CCD, very approximately by the Y-binning factor.
For example, when compared to a 1392 x 1040 unbinned image, an image captured with 3 (X) x 2 (Y) binning would have a resolution of 464 x 520, a signal amplitude six times higher, and would read out in about half the time.

SoftBin:
The Software Binning is very similar to the On-chip Binning except that the sum of the pixel signals is not limited by the CCD saturation level. The "SoftBin" algorithm is slower than the On-chip binning.

SubArea:
Only the image data for this rectangular sub-area being transferred to the PC. This feature can be used in conjunction with binning, to provide the user with considerable flexibility in image location, size, frame rate, and resolution.
The time taken to read out the CCD will be reduced, in proportion to the number of whole TV lines that are above and below the sub-area selected. For example, if the sub-area (200,200) - (700, 700) were selected, then readout would occur in about half the time of a full 1392 x 1040 image, and the image returned to the PC would have resolution 500 x 500. If in addition binning of 2 x 5 were selected, the sub-area selected would be unchanged but the image resolution would become 250 x 100.

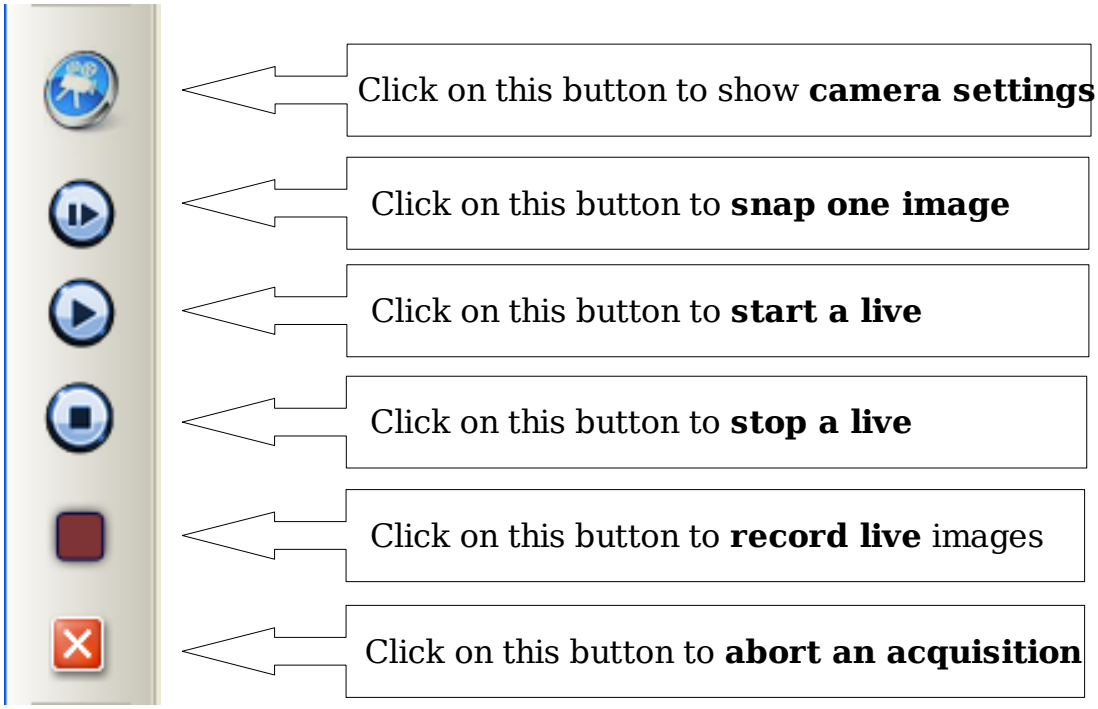
Exposure:
The integration or exposure period of the CCD is user-selectable via software over a very large range. The integration period is defined by a 14 bit number (range 1 to 16384) together with a unit (microseconds, milliseconds, or seconds). Very fine control of integration period is therefore available to the user.

Note that the driver written by Photonic Science always uses the finest unit available for any particular selected integration period, eg. an integration period of 17 seconds is transmitted to the camera as period = 17, unit = seconds, whereas an integration period of 16 seconds is transmitted to the camera as period = 16000, unit = milliseconds.

Miscellaneous:

- **FrameTime (ms)**: This option directly affect the number of frames displayed per second. The *"FrameTime"* is the time between two consecutive acquisition triggering signal. If the FrameTime is longer than the exposure time then the camera will wait a certain quantity of time before starting the next acquisition. If it is shorter, then the camera will start the next acquisition as fast as possible.

2) Image acquisition



Note: You can stop recording by clicking again on the record button or clicking on the stop button. If you confirm the image sequence recording, then a dialog box will ask you to select/create a folder to store all frames of the sequence. The frames are saved in the folder as "xxxx.tiff".

4. File images

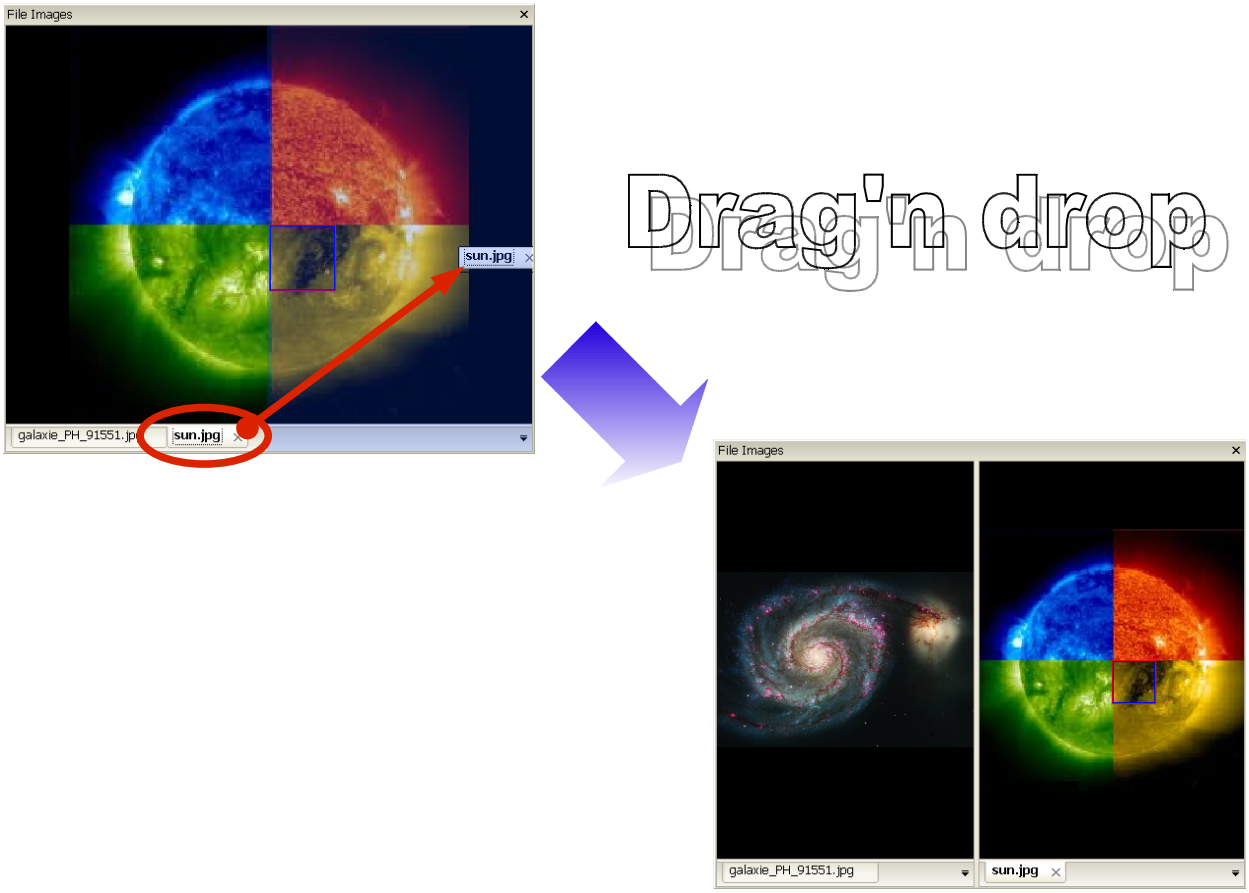
1) Open / Save

To open a image file, in the menu bar click on *"File->Open..."*. Then a File dialog box allow you to browse for your file. You can open multiple images in once using "CTRL" or "SHIFT" buttons on your keyboard while selecting files in the browser.

To save an image, go in the menu bar and click on *"File->SaveAs..."*. Then a File dialog box allow you to select the location for saving the image. Make sure the active display correspond to the image you want to save. The active display is the last Focused Image. To set the focus on an image just click on it.

2) Managing files and panels

All file images are stored in the “File Images” note-book. You can navigate between images by clicking on the image page Tabs or on the little arrow on the right.



5. Remote control

1) Socket and server keywords

=> socket.socket(socket.**AF_INET**, socket.**SOCK_STREAM**)

GetAvailableCameras: return available camera names

GetCurrentCam: return current camera name

Open;CameraName: open the camera named *CameraName*;
ex: *'Open;ImageStar9000'*
ex: *'Open;QuadIS9000'*

Close: close the current camera

Abort: abort the current camera exposition

SetAutoSave: *enable/disable the image saving after each "Snap".*
Save path is 'FileDirectory\\FileName_FileNumber.FileFormat'

SetFileDirectory: set the current saving file directory

SetFileName: the current saving file name

SetFileNumber: set the current saving file number

SetFileFormat: set the current saving file format

GetFilePath: return the current save path.
FileDirectory\\FileName_FileNumber.FileFormat

SetTagTxt: Add some text to the 'DESCRIPTION' TIFF tag of the image file (while saving using 'AutoSave')

Example: How to acquire and save my images
>>> 'SetFileDirectory;MyDirectory'
>>> 'SetFileName;MyFileName'
>>> 'SetFileNumber;0'
>>> 'SetFileFormat;tiff'
>>> 'SetAutoSave;1'
>>> 'SetTagTxt;this my tiff comment tag'
>>> 'Snap'
>>> 'Snap'
>>> ...
As soon as 'Snap' has returned, you can send a new 'Snap' command
All post process and storage to disk is done in parallel of the next snap
Between two 'Snap' you can change the Tiff tag or the file path, name and number
Note: The file number is automatically incremented if not using 'SetFileNumber;x'

GetImage: return *(width,height,data_length,compression_mode)* then the image data as a String. Loop on *'Socket.recv()'* until the received data length is equal to *'data_length'.*

GetMode: return the current camera mode

GetSize: return the current camera size

GetSizeMax: return the current camera maximum size

GetName: return the current camera name

GetStatus: return the current camera acquisition status; *False* if not acquiring

SetExpoMS: Set exposure in Millisec; ex: *'SetExpoMS;100'*

GetOptions: return all the current camera options

Get*...* : return the current value of a camera option

Set*...* : set a camera option value; refer to *'GetOptions'* for camera option names

ex: *'SetBinning;(2,2)' or 'GetBinning'*
ex: *'SetExposure;(100,'Second')'* or *'GetExposure'*
ex: *'SetVideoGain;30'* or *'GetVideoGain'*
ex: *'SetFileDirectory;C:\\MyFrames'* or *'GetFileDirectory'*

Note: for MultiCameras you can add *';camera_id'* to commands to drive one specific sub-camera

ex: *'SetBinning;(2,2);1'* to set the binning of the sub-camera number 1
ex: *'GetBinning;2'* to get the current binning value of the sub-camera 2

2) Python example

```
import socket

IP = “162.183.1.24”
PORT = 50000

def SendAndRecv(cmd):
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    sock.connect((IP, PORT))
    sock.send(cmd+'\n')

    if cmd == "GetImage":
        nx,ny,data_len = sock.recv(1024).split(';')
        nx,ny,data_len = int(nx),int(ny),int(data_len)

        data = ""
        while 1:
            rep = sock.recv(data_len)
            data = "".join([data,rep])
            if len(data)>=data_len:
                break

        data = ((nx,ny),data)

    else:
        data = sock.recv(10240)
        sock.close()

    return data
```